CLAIMS IN THE CASE

Please amend Claims 7 and 12, as follows.

Please add Claim 16.

A2 Sub B1

Claim 1. (Original) A method of transferring between types of conversion processes in a computer which converts instructions from a target instruction set to a host instruction set comprising the steps of:

executing code morphing software including an interpreter and a translator to generate host instructions from target instructions,

detecting at intervals whether the interpreter or the translator is operating,

increasing a count if the interpreter is operating and decreasing the count if the translator is operating, and

changing from interpreting to translating a sequence of target instructions when the count reaches a selected maximum.

Claim 2 (Original) A method as claimed in Claim 1 in which the interval is a selected time period.

Claim 3. (Original) A method as claimed in Claim 1 in which the interval is a selected number of executed target instructions.

Claim 4. (Original) A method as claimed in Claim 1 in which the amount the count is increased at a detection of interpretation is selectable.

Claim 5. (Original) A method as claimed in Claim 1 in which the amount the count is decreased at a detection of translation is selectable.

Claim 6. (Original) A method as claimed in Claim 1 comprising the further steps of:

counting each instance in which a sequence of instructions is interpreted,

changing from interpreting to translating a sequence of target instructions when the count of instances reaches a selected maximum.

Claim 7. (Currently Amended) A method as claimed in Claim ~~7~~ 1comprising the further steps of:

gathering statistics regarding each sequence of instructions, and

optimizing translation of a sequence of instructions based on statistics gathered.

Claim 8. (Original) A method as claimed in Claim 1 in which the step of changing from interpreting to translating a sequence of target instructions when the count reaches a selected maximum includes translation with limited optimization, and

which further includes the steps of:

testing while executing a sequence of target instructions translated with limited optimization to determine whether the sequence should be further optimized, and

retranslating and further optimizing in response to the testing.

Claim 9. (Original) A method as claimed in Claim 8 in which the step of testing while executing a sequence of target instructions translated with limited optimization includes counting each instance in which a sequence of instructions is executed, and

the step of retranslating and further optimizing occurs when the count of instances reaches a selected maximum.

Claim 10. (Original) A method of optimizing execution by a computer which dynamically converts instructions from a target instruction set to a host instruction set comprising the steps of:

providing a plurality of instruction conversion processes each providing a different level of optimization for converted instructions from a target instruction set to a host instruction set,

providing means for determining dynamically which conversion process best converts each sequence of instructions, and

converting a sequence of instructions using a conversion process determined to best convert the sequence of instructions.

Claim 11. (Original) A method as claimed in Claim 10 in which the conversion processes include interpretation and translation.

Claim 12. (Currently Amended) A method as claimed in Claim 10 in which the conversion processes include interpretation, translation with minimal optimization, translation with advanced optimization.[[.]]

Claim 13. (Original) A method as claimed in Claim 10 in which the means for determining dynamically which conversion process best converts each sequence of instructions depends on the number of times each sequence is converted by a particular conversion process.

Claim 14. (Original) A method as claimed in Claim 10 in which the means for determining dynamically which conversion process best converts each sequence of instructions depends on a ratio of the number of times one conversion process is run compared to another conversion process.

Claim 15. (Original) A method as claimed in Claim 10 in which the means for determining dynamically which conversion process best converts each sequence of instructions

depends on the number of times each sequence is converted by a particular conversion process, and

depends on a ratio of the number of times one conversion process is run compared to another conversion process.

Claim 16. (New) A method of transferring between types of conversion processes in a computer which converts instructions from a target instruction set to a host instruction set comprising the steps of:

executing code morphing software including an interpreter and a translator to generate host instructions from target instructions,

comparing interpreter usage to translator usage when executing said code morphing software to produce an interpreter usage to translator usage factor, and

changing from interpreting to translating a sequence of target instructions if the interpreter usage to translator usage factor crosses a threshold.